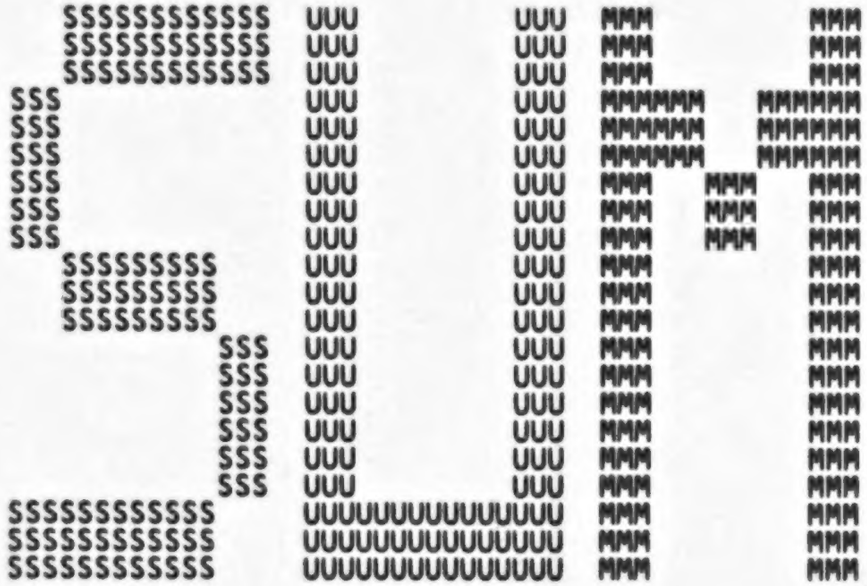
SSSSSSSSSSS UUU UUU MMM MMM
SSSSSSSSSSS UUU UUU MMM MMM
SSSSSSSSSSS UUU UUU MMM MMM
SSS UUU UUU MMMMMM MMMMM
SSS UUU UUU MMMMMM MMMMM
SSS UUU UUU MMMMMM MMMMM
SSS UUU UUU MMM MMM MMM
SSS UUU UUU MMM MMM MMM
SSS UUU UUU MMM MMM MMM
SSSSSSSSS UUU UUU MMM MMM
SSSSSSSSS UUU UUU MMM MMM
SSSSSSSSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSS UUU UUU MMM MMM
SSSSSSSSSSS UUUUUUUUUUUUUUUU MMM MMM
SSSSSSSSSSS UUUUUUUUUUUUUUUU MMM MMM
SSSSSSSSSSS UUUUUUUUUUUUUUUU MMM MMM

FILEID**SUMOPEN

```
  SSSSSSSS  UU      UU  MM      MM    OOOOOO    PPPPPPPP    EEEEEEEEEE  NN      NN
  SSSSSSSS  UU      UU  MM      MM    OOOOOO    PPPPPPPP    EEEEEEEEEE  NN      NN
SS          UU      UU  MMMM  MMMM  OO      OO  PP      PP  EE          NN      NN
SS          UU      UU  MMMM  MMMM  OO      OO  PP      PP  EE          NN      NN
SS          UU      UU  MM  MM  MM  OO      OO  PP      PP  EE          NNNN    NN
SS          UU      UU  MM  MM  MM  OO      OO  PP      PP  EE          NNNN    NN
  SSSSSS    UU      UU  MM      MM  OO      OO  PPPPPPPP    EEEEEEEE    NN  NN  NN
  SSSSSS    UU      UU  MM      MM  OO      OO  PPPPPPPP    EEEEEEEE    NN  NN  NN
        SS  UU      UU  MM      MM  OO      OO  PP          EE          NN    NNNN
        SS  UU      UU  MM      MM  OO      OO  PP          EE          NN    NNNN
        SS  UU      UU  MM      MM  OO      OO  PP          EE          NN      NN    ....
        SS  UU      UU  MM      MM  OO      OO  PP          EE          NN      NN    ....
SSSSSSSS    UUUUUUUUUU  MM      MM    OOOOOO    PP          EEEEEEEEEE  NN      NN    ....
SSSSSSSS    UUUUUUUUUU  MM      MM    OOOOOO    PP          EEEEEEEEEE  NN      NN    ....


LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  SUM$OPEN
0000     2
0000     3          .IDENT  'V04-000'
0000     4
0000     5  ;
0000     6  ;*****************************************************************************
0000     7  ;*                                                                           *
0000     8  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     9  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    10  ;*  ALL RIGHTS RESERVED.                                                     *
0000    11  ;*                                                                           *
0000    12  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    13  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    14  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    15  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    16  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    17  ;*  TRANSFERRED.                                                              *
0000    18  ;*                                                                           *
0000    19  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    20  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    21  ;*  CORPORATION.                                                              *
0000    22  ;*                                                                           *
0000    23  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    24  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
0000    25  ;*                                                                           *
0000    26  ;*                                                                           *
0000    27  ;*****************************************************************************
0000    28  ;
0000    29
0000    30  ;++
0000    31  ; FACILITY:
0000    32  ;
0000    33  ; ABSTRACT:
0000    34  ;
0000    35  ;       Source Update Merge procedure to check the existance of
0000    36  ;       the files in an update files list.
0000    37  ;
0000    38  ;
0000    39  ;
0000    40  ; ENVIRONMENT:  USER MODE
0000    41  ;
0000    42  ; AUTHOR:       R. Newland
0000    43
0000    44  ;
0000    45  ; MODIFIED BY:
0000    46  ;
0000    47  ;       V02-001         BLS0139         Benn Schreiber          26-Jan-1981
0000    48  ;                Remove deallocation of unused part of resultant name buffer
0000    49  ;
0000    50  ;--
```

```
                     0000     52          .SBTTL  DECLARATIONS
                     0000     53 ;
                     0000     54 ;
                     0000     55 ; Macro definitions
                     0000     56 ;
                     0000     57          DEFUPFBLK                       ; Source update merge
                     0000     58          $FABDEF                         ; FAB block
                     0000     59          $NAMDEF                         ; NAM block
                     0000     60 ;
                     0000     61 ;
                     0000     62 ;
                     0000     63 ;
                     0000     64 ; Local storage
                     0000     65 ;
                     0000     66 ;
                 00000000     67          .PSECT  SUM$RO_DATA,NOEXE,NOWRT,LONG
                     0000     68 ;
                     0000     69 ;
                     0000     70 SUM_DEFNAME:
 44 50 55 2E 00'     0000     71          .ASCIC  /.UPD/                  ; Default file name
                04   0000
                     0005     72 ;
                     0005     73 SUM_MAXRSS:
             00000100  0005     74          .LONG   256                     ; Max size of resultant file name string
                     0009     75 ;
                     0009     76 ;
                     0009     77 ;
                 00000000     78          .PSECT  SUM$RW_DATA,NOEXE,LONG
                     0000     79 ;
```

```
                                        .SBTTL  SUM$OPEN

                                ;
                                ;++
                                ; Functional description:
                                ;
                                ;       This procedure determines which files in an update files list
                                ;       can be successfully opened.  The NAM block information of each
                                ;       file which can be opened is saved so that subsequent opens can
                                ;       be performed quickly.  The resultant file specification string
                                ;       is also saved.  For files that can not be opened because of a
                                ;       'file not found' error a warning message is issued and the file
                                ;       entry removed.  For files that can not be opened for any other
                                ;       reason a fatal error message is issued.
                                ;
                                ;       The NAM block of the associated source file is used to
                                ;       supply initial default information.
                                ;
                                ;       The input parameter, address of first update file, will be
                                ;       changed if the first file (or the new first file) can not be
                                ;       found.  If all files can not be found the pointer will become zero.
                                ;
                                ;
                                ; Calling sequence:
                                ;
                                ;       CALLG/CALLS
                                ;
                                ;
                                ; Inputs:
                                ;
                                ;        4(AP)  = Source file NAM block address
                                ;        8(AP)  = Pointer to address of first entry in Update list
                                ;
                                ; Outputs:
                                ;
                                ;       None
                                ;;
                                ;
                                ;--
                                ;
                00000000        .PSECT  SUM$CODE,NOWRT,LONG
                                ;
                                .DEBUG  SUM$OPEN
                                ;
                0C1C            .ENTRY  SUM$OPEN,^M<R2,R3,R4,R10,R11>
                                ;
   52   00000000'EF   9E        MOVAB   SUM$FAB,R2                      ; Set FAB pointer
35 A2   00000000'EF   90        MOVB    SUM_DEFNAME,FAB$B_DNS(R2) ; Initialise default file name
30 A2   00000001'EF   DE        MOVAL   SUM_DEFNAME+1,FAB$L_DNA(R2)
     00 04 A2   18   E5        BBCC    #FAB$V_NAM,FAB$L_FOP(R2),10$ ; Set for open by file name
                                10$:
      53   04 AC   D0        MOVL    4(AP),R3                        ; Set related NAM block pointer
      5A   08 BC   D0        MOVL    @8(AP),R10                      ; and Update file block pointer
      5B   08 BC   DE        MOVAL   @8(AP),R11                      ; Set pointer to previous block
                                20$:
      54   38 AA   9E        MOVAB   UPF_T_NAM(R10),R4               ; Set NAM block pointer
   34 A2   20 AA   90        MOVB    UPF_Q_CMNT(R10), -              ; Set file name string size and address
                                        FAB$B_FNS(R2)
```

```
     2C A2    24 AA    D0  0033          MOVL    UPF_Q_CMNT+4(R10), -
                           0038                  FAB$L_FNA(R2)
     28 A2       54    D0  0038          MOVL    R4,FAB$L_NAM(R2)          ; Set NAM block pointer
     10 A4       53    D0  003C          MOVL    R3,NAM$L_RLF(R4)          ; Set related NAM block pointer
     0A A4    FF 8F    90  0040          MOVB    #255,NAM$B_ESS(R4)        ; Set size of expanded string
     02 A4    FF 8F    90  0045          MOVB    #255,NAM$B_RSS(R4)        ; Set size of resultant string buffer
              04 A4    DF  004A          PUSHAL  NAM$L_RSA(R4)             ; Address to store resultant string
          00000005'EF  DF  004D          PUSHAL  SUM_MAXRSS                ; Address of max resultant string length
 00000000'GF     02    FB  0053          CALLS   #2,G^LIB$GET_VM           ; Get memory block for resultant string
          6E     50    E9  005A          BLBC    R0,50$                    ; Error if LBC
     0C A4    04 A4    D0  005D          MOVL    NAM$L_RSA(R4),NAM$L_ESA(R4) ; Set expanded string buffer
                           0062          $OPEN   FAB=R2,ERR=SUM$OPEN_ERR   ; Open file
          3F     50    E8  0071          BLBS    R0,40$                    ; OK if LBS
 00000000'8F     50    D1  0074          CMPL    R0,#RMS$_FNF              ; Was error 'file not found'?
                 03    13  007B          BEQL    30$                       ; Yes if EQL
               0060    31  007D          BRW     90$
                           0080  30$:
              04 A4    DF  0080          PUSHAL  NAM$L_RSA(R4)             ; Return resultant file spec block
          00000005'EF  DF  0083          PUSHAL  SUM_MAXRSS
 00000000'GF     02    FB  0089          CALLS   #2,G^LIB$FREE_VM
          46     50    E9  0090          BLBC    R0,80$                    ; Error if LBC
               004B    30  0093          BSBW    RETURN_FNS                ; Return file name string block
          40     50    E9  0096          BLBC    R0,80$                    ; Error if LBC
                 6B    DF  0099          PUSHAL  (R11)                     ; Return the update block
          00000000'EF  DF  009B          PUSHAL  SUM$BLSZE                 ; Size of block
          5A     6A    D0  00A1          MOVL    (R10),R10                 ; Advance pointer before returning block
 00000000'GF     02    FB  00A4          CALLS   #2,G^LIB$FREE_VM          ; Free memory
          2B     50    E9  00AB          BLBC    R0,80$                    ; Error if LBC
          6B     5A    D0  00AE          MOVL    R10,(R11)                 ; Relink update blocks
                 21    11  00B1          BRB     70$
                           00B3  40$:
                           00B3          $CLOSE  FAB=R2,ERR=SUM$CLOSE_ERR  ; Close file
          1B     50    E9  00C2          BLBC    R0,90$                    ; Error if LBC
          53     54    D0  00C5          MOVL    R4,R3                     ; Next related NAM block will be
                           00C8                                            ; this files' NAM block
               0016    30  00C8          BSBW    RETURN_FNS                ; Return file name string block
                           00CB  50$:
          0B     50    E9  00CB          BLBC    R0,80$                    ; Error if LBC
                           00CE  ;
          5B     5A    D0  00CE          MOVL    R10,R11                   ; Form previous block pointer
          5A     6A    D0  00D1          MOVL    (R10),R10                 ; Point to next Update file block
                           00D4  70$:
                 0A    13  00D4          BEQL    90$                       ; and finish if end of list
               FF51    31  00D6          BRW     20$                       ; Go back for next file
                           00D9  80$:
 00000000'EF     00    FB  00D9          CALLS   #0,SUM$LIB_ERR            ; Report library error
                           00E0  90$:
                       04  00E0          RET
```

```
                        00E1   187          .SBTTL  RETURN_FNS
                        00E1   188 ;
                        00E1   189 ;
                        00E1   190 ; Subroutine to return file name string block to virtual memory
                        00E1   191 ;
                        00E1   192 ; Inputs:
                        00E1   193 ;
                        00E1   194 ;       R10 = Address of SUM block
                        00E1   195 ;
                        00E1   196 ; Outputs:
                        00E1   197 ;
                        00E1   198 ;       R0  = Success/error status
                        00E1   199 ;
                        00E1   200 ;
                        00E1   201 RETURN_FNS:
              20 AA   B5 00E1   202          TSTW    UPF_Q_CMNT(R10)         ; Was there a file name string?
                 0D   13 00E4   203          BEQL    10$                     ; No if EQL so no memory block to return
              24 AA   DF 00E6   204          PUSHAL  UPF_Q_CMNT+4(R10)       ; Address of file name string
              20 AA   DF 00E9   205          PUSHAL  UPF_Q_CMNT(R10)         ; Size of string
00000000'GF     02   FB 00EC   206          CALLS   #2,G^LIB$FREE_VM                ; Return memory
                        00F3   207 10$:
                     05 00F3   208          RSB
                        00F4   209 ;
                        00F4   210 ;
                        00F4   211          .END
```

```
$$.TMP1                         =  00000002
$$.TMP2                         =  00000052
BIT...                          =  00000005
FAB$B_DNS                       =  00000035
FAB$B_FNS                       =  00000034
FAB$L_DNA                       =  00000030
FAB$L_FNA                       =  0000002C
FAB$L_FOP                       =  00000004
FAB$L_NAM                       =  00000028
FAB$V_NAM                       =  00000018
LIB$FREE_VM                        ********    X   04
LIB$GET_VM                         ********    X   04
NAM$B_ESS                       =  0000000A
NAM$B_RSS                       =  00000002
NAM$K_BLN                       =  00000060
NAM$L_ESA                       =  0000000C
NAM$L_RLF                       =  00000010
NAM$L_RSA                       =  00000004
RETURN_FNS                         000000E1 R       04
RMS$_FNF                           ********    X   04
SIZ...                          =  00000001
SUM$BLSZE                          ********    X   04
SUM$CLOSE_ERR                      ********    X   04
SUM$FAB                            ********    X   04
SUM$LIB_ERR                        ********    X   04
SUM$OPEN                           00000000 RG  D   04
SUM$OPEN_ERR                       ********    X   04
SUM_DEFNAME                        00000000 R       02
SUM_MAXRSS                         00000005 R       02
SUM_M_AUDIT                     =  00000001
SUM_M_AUDITNEW                  =  00000002
SUM_M_DELETE                    =  00000010
SUM_M_SRCUPD                    =  00000004
SUM_M_SUBCLSH                   =  00000008
SUM_V_AUDIT                     =  00000000
SUM_V_AUDITNEW                  =  00000001
SUM_V_DELETE                    =  00000004
SUM_V_SRCUPD                    =  00000002
SUM_V_SUBCLSH                   =  00000003
SYS$CLOSE                          ********  GX    04
SYS$OPEN                           ********  GX    04
UPF_B_EDFLAGS                      00000009
UPF_B_FIFLAGS                      00000008
UPF_B_FILENO                       0000000C
UPF_K_BLN                          00000098
UPF_L_PTR                          00000000
UPF_Q_AUDDS                        00000018
UPF_Q_CMNT                         00000020
UPF_Q_EDITS                        00000010
UPF_T_AUDST                        00000028
UPF_T_NAM                          00000038
UPF_W_DOT                          0000000A
UPF_W_LOC1                         00000004
UPF_W_LOC2                         00000006
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000098 (152.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SUM$RO_DATA	00000009 (9.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
SUM$RW_DATA	00000000 (0.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SUM$CODE	000000F4 (244.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.31
Command processing	105	00:00:00.53	00:00:02.22
Pass 1	194	00:00:04.29	00:00:09.68
Symbol table sort	0	00:00:00.40	00:00:00.89
Pass 2	55	00:00:00.87	00:00:02.16
Symbol table output	7	00:00:00.07	00:00:00.25
Psect synopsis output	2	00:00:00.03	00:00:00.11
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	394	00:00:06.29	00:00:15.63

The working set limit was 1200 pages.
21065 bytes (42 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 379 non-local and 9 local symbols.
211 source lines were read in Pass 1, producing 20 object records in Pass 2.
17 pages of virtual memory were used to define 15 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SUM.OBJ]SUM.MLB;1	2
_255DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	11

492 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SUMOPEN/OBJ=OBJ$:SUMOPEN MSRC$:SUMOPEN/UPDATE=(ENH$:SUMOPEN)+LIB$:SUM/LIB

SUMLIST
LIS

SUMMAIN
LIS

SUMTFRVEC
LIS

SUMFILES
LIS

SUMMSG
LIS

SUMERROR
LIS

SUMOPEN
LIS

SYS

UPDATE
LIS

SYS
MAP